Step-by-Step Guide to Building a Data Pipeline in Azure Data Factory

In this guide, you'll build a data pipeline using **Azure Data Factory (ADF)** that copies data from a **student-dataset.csv** file stored in **Azure Blob Storage** to an **Azure SQL Database**. After the data transfer, you'll query the SQL database to display the **Student Count by Country**.

Prerequisites:

1. **Azure Subscription**: Select Free Trial.
2. **Azure Storage Account**: For storing the **student-dataset.csv** file.
3. **Azure SQL Database**: To store and query the student data.
4. **Azure Data Factory**: To create and manage the data pipeline.
5. SQL Server Management Studio (SSMS) or Query editor (preview): To query and the database.

Summary of Steps:

1. Set up an Azure Storage account and upload the student-dataset.csv file.
2. Set up an Azure SQL Database and create the Students table.
3. Create an Azure Data Factory instance.
4. Set up linked services for Azure Blob Storage and Azure SQL Database.
5. Create datasets for the CSV file and the SQL table.
6. Build the pipeline using a **Copy Data** activity to transfer the data.
7. Test, publish, and verify the data transfer.
8. Query the SQL database for the **Student Count by Country**.

PROJECT 1: Build an End-to-end data pipeline **using Azure Data Factory**

Objective:

The goal of this project is to build an end-to-end data pipeline that extracts student data from a CSV file, processes the data, stores it in Azure SQL Database. You will leverage Azure services such as Azure Storage, Azure Data Factory, Azure SQL Database. See the attached **Step-by-Step Guide to Building a Data Pipeline in Azure Data Factory**

Step 1: Upload student-dataset.csv to Azure Blob Storage

1.1 Create a Storage Account:

- In the **Azure Portal**, search for **Storage accounts** and click **Create**.
- Select your subscription (Free Trial) and resource group (Create or select an existing one).
- Give the storage account a unique name and select your region.
- Choose **Locally-redundant storage (LRS)** for redundancy and click **Review + Create**.

1.2 Create a Blob Container:

- Once your storage account is deployed, go to it and click **Containers** under **Data storage**.
- Click **+ Container**, name it (e.g., student-data), and set the access level to **Private**.

1.3 Upload student-dataset.csv:

- Download CSV file with student data:
- **`student-dataset.csv`** `(first few lines)`

  ```
  StudentID,Name,Country,City,Latitude,Longitude,Gender,Ethnic.group,Age,English.grade,Math.grade,Sciences.grade,Language.grade,Portfolio.rating,Coverletter.rating,Refletter.rating
  0,Kiana Lor,China,Suzhou,31.31,120.62,F,NA,22,3.5,3.7,3.1,1,4,4,4
  1,Joshua Lonaker,United States of America,Santa Clarita,34.39,-118.54,M,NA,22,2.9,3.2,3.6,5,5,4,5
  2,Dakota Blanco,United States of America,Oakland,37.8,-122.27,F,NA,22,3.9,3.8,3.2,5,3,3,4
  ```
- In your container, click **Upload**, select your student-dataset.csv file, and click **Upload**.

Step 2: Create an Azure SQL Database

2.1 Create a SQL Database:

- In the **Azure Portal**, search for **SQL databases** and click **Create**.
- Choose your subscription, resource group, and create a new SQL server with admin credentials.
- Select the **Free or Basic** pricing tier (suitable for small projects).
- Review the settings and click **Create**.

2.2 Configure SQL Firewall:

- After the database is deployed, go to the SQL server resource.
- Under **Security**, click **Firewalls and virtual networks**.
- Add the IP addresses or ranges that should access the database (e.g., your local IP).
- Click **Save**.

2.3 Create the Students Table:

- Use **SQL Server Management Studio (SSMS)** or **Azure Data Studio** to connect to your SQL Database.
- Run the following SQL query to create the Students table. Here are the minimum number of columns:

  ```
  SQL Code:
  CREATE TABLE Students (
      StudentID INT PRIMARY KEY,
      Name NVARCHAR(100),
      Country NVARCHAR(100),
      …
  );
  ```

Step 3: Set Up Azure Data Factory

3.1 Create an Azure Data Factory Instance:

- In the **Azure Portal**, search for **Data Factories** and click **Create**.
- Select your subscription, resource group, and provide a unique name for the Data Factory.
- Choose your region and click **Create**.

3.2 Launch Data Factory Studio:

- After deployment, click **Author & Monitor** to open Data Factory Studio.

Step 4: Create Linked Services in Azure Data Factory

4.1 Create a Linked Service for Blob Storage:

- In Data Factory Studio, go to the **Manage** tab (gear icon on the left).
- Under **Connections**, click **Linked Services** and **+ New**.
- Select **Azure Blob Storage** and configure:
  - Name: BlobStorageLinkedService
  - Authentication type: **Account key**
  - Select your storage account and test the connection.
  - Click **Create**.

4.2 Create a Linked Service for Azure SQL Database:

- In the **Linked Services** section, click **+ New** again.
- Select **Azure SQL Database** and configure:
  - Name: AzureSQLLinkedService
  - Server name, database name, and the credentials you used when creating the SQL Database.
  - Test the connection and click **Create**.

Step 5: Create Datasets

5.1 Create a Dataset for the CSV File:

- Go to the **Author** tab (pencil icon), right-click **Datasets**, and select **New Dataset**.
- Select **Azure Blob Storage** and choose **DelimitedText** as the format.
- Configure:
  - Name: StudentCSV
  - Linked Service: Select BlobStorageLinkedService.
  - File Path: Browse and select the **student-dataset.csv** file.
  - First Row as Header: **True**.
  - Schema: Import the schema from the file.
  - Click **OK**.

5.2 Create a Dataset for the SQL Table:

- In the **Datasets** section, create another dataset by selecting **Azure SQL Database**.
- Configure:
  - Name: SQLStudentTable
  - Linked Service: Select AzureSQLLinkedService.
  - Table: Select the **Students** table you created earlier.
  - Click **OK**.

Step 6: Create the Data Pipeline

6.1 Create a New Pipeline:

- In the **Author** tab, right-click **Pipelines** and select **New Pipeline**.
- Name the pipeline (e.g., CopyStudentDataPipeline).

6.2 Add Copy Data Activity:

- Drag and drop the **Copy Data** activity from the activities pane into the pipeline.
- Click on the activity, then configure the **Source**:
  - Source Dataset: StudentCSV
  - Optional: You can add **column mappings** if the schema of the CSV differs from the SQL table.

6.3 Configure the Sink (Destination):

- Click on the **Sink** tab:
  - Sink Dataset: SQLStudentTable.

6.4 Test the Pipeline:

- Click **Debug** to run the pipeline.
- After the pipeline runs, check the **Output** tab for logs to ensure the data has been copied successfully.

6.5 Query the Data in Azure SQL Database

- Use **SSMS** or **Azure Data Studio** to connect to the SQL Database and run the following query to display the **Student Count by Country**:

  ```
  SQL code:
  SELECT Country, COUNT(*) AS StudentCount
  FROM Students
  GROUP BY Country;
  ```

6.6 Publish the Pipeline

Once the pipeline works successfully during the debug, you can publish it:

- Click **Publish All** in Data Factory Studio to save the pipeline for future use.
- Schedule the pipeline to run automatically or trigger it manually as needed.

Conclusion:

You've now successfully built an Azure Data Factory pipeline that:

- Copies data from **Azure Blob Storage** (student-dataset.csv) to an **Azure SQL Database**.
- Allows you to query and display the **Student Count by Country** from the SQL Database.

This pipeline can be expanded or customized to handle larger datasets or additional transformations as needed.